

Section 7: Simple SQL Exercises

How many records are in the nyc_streets table?

How many streets in New York have names that start with 'B'?

What is the population of New York city?

The "nyc_census_blocks" table includes a "popn_total" field which is the number of people in each block.

What is the population of the Bronx?

The "nyc_census_blocks" table includes a "boroname" field which is the borough each block is in. 'The Bronx' is a borough.

How many "neighborhoods" are in each borough?

The "nyc_neighborhoods" table includes a "boroname" field and a "name" field. Group by!

For each borough, what percentage of the population is white?

The "nyc_census_blocks" table includes a "popn_white" field that is the number of self-identified white people in each block.

Data Dictionary

Useful columns in the “nyc_census_blocks” table:

blkid	A 15-digit code that uniquely identifies every census block. Eg: 360050001009000
popn_total	Total number of people in the census block
popn_white	Number of people self-identifying as “white” in the block
popn_black	Number of people self-identifying as “black” in the block
popn_nativ	Number of people self-identifying as “native american” in the block
popn_asian	Number of people self-identifying as “asian” in the block
popn_other	Number of people self-identifying with other categories in the block
hous_total	Number of housing units in the block
hous_own	Number of owner-occupied housing units in the block
hous_rent	Number of renter-occupied housing units in the block
boroname	Name of the New York borough: Manhattan, The Bronx, Brooklyn, Staten Island, Queens
the_geom	MultiPolygon boundary of the block

Useful Functions

Count(*)	An aggregate function that returns the number of records in the query.
GROUP BY [field]	A SQL statement that performs aggregations based on the distinct values of the provided field.
[field] LIKE 'V%'	A SQL pattern matching operator, where the field is matched against a pattern and '%' is the 'global match' character. The example returns all field values starting with 'V'.
Sum([field])	An aggregate function that returns the total value of the field over all records in the query.
Avg([field])	An aggregate function that returns the average value of the field over all records in the query.

Section 9: Geometry Exercises

What is the area of the ‘West Village’ neighborhood?

What are the units of your answer?

The “nyc_neighborhoods” table includes a “name” field.

What is the first coordinate in ‘Pelham St’ in well-known text?

What are the units of your answer?

The “nyc_streets” table includes a “name” field.

What is the GML representation of the ‘Broad St’ subway station?

What about the KML? Why are they different?

The “nyc_subway_stations” table includes a “name” field.

How many census blocks in New York City have a hole in them?

How do you represent polygons with a hole?

What is the most westerly subway station?

Try using “ORDER” and “LIMIT”.

What is the area of Manhattan in acres?

The “nyc_census_blocks” table is an area geometry and has a “boroname” column.
There are 4047 square meters in an acre.

Data Dictionary

Useful columns in the “nyc_streets” table:

name	Common name of the street
oneway	Is this a one-way street?
type	What kind of street is this?
the_geom	MultiLinestring geometry of the street

Useful columns in the “nyc_subway_stations” table:

name	Common name of the station
routes	Comma-separated list of routes that serve this station
the_geom	Point geometry of the street

Useful Functions

Sum([field])	An aggregate function that returns the total value of the field over all records in the query.
LIMIT n	Restrict the query to return only the first “n” rows.
ORDER BY [field]	Return the query in order sorted by the field.
ORDER BY [field] DESC ORDER BY [field] ASC	Return the query in descending/ascending order sorted by the field.
ST_X(point)	Returns the X coordinate of the point
ST_Y(point)	Returns the Y coordinate of the point
ST_Length(geometry)	Returns the length of the geometry
ST_Area(geometry)	Returns the area of the geometry
ST_StartPoint(line)	Returns the first point in the line
ST_EndPoint(line)	Returns the last point in the line
ST_NPoints(geometry)	Returns the number of vertices in any geometry
ST_NumInteriorRings(polygon)	Returns the number of interior rings (holes) in a polygon
ST_NumGeometries(collection)	Returns the number of sub-geometries in any geometry collection (MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRYCOLLECTION)
ST_GeometryN(geometry, n)	Returns the n’t h geometry in the collection (starting from 1)
ST_AsGML(geometry)	Returns the GML representation
ST_AsKML(geometry)	Returns the KML representation
ST_AsGeoJSON(geometry)	Returns the GeoJSON representation
ST_AsText(geometry)	Returns the well-known-text representation

Section 11: Spatial Relationship Exercises

What is the well-known text for the street 'Atlantic Commons'?

What neighborhood is `LINESTRING(586782 4504202,586864 4504216)` in?

How many people live within 50 meters of `POINT(586782 4504202)`?

The "nyc_census_blocks" table includes a "popn_total" field.

For '`LINESTRING(0 0, 2 2)`' and '`POINT(1 1)`' which of these relationships are true? Intersects, Touches, Contains, Disjoint, Overlaps, Crosses, Within.

A sub-query might make this terser to type.

How far apart are 'Columbus Cir' and 'Fulton Ave'?

For SQL experts only!

Useful Functions

Sum([field])	An aggregate function that Returns the total value of the field over all records in the query.
ST_Contains(A, B)	Returns true if geometry A contains geometry B
ST_Crosses(A, B)	Returns true if geometry A crosses geometry B
ST_Disjoint(A, B)	Returns true if the geometries do not “spatially intersect”
ST_Distance(A, B)	Returns the minimum distance between geometry A and geometry B
ST_DWithin(A, B, d)	Returns true if geometry A is distance or less from geometry B
ST_Equals(A, B)	Returns true if geometry A is the same as geometry B
ST_Intersects(A, B)	Returns true if geometry A intersects geometry B
ST_Overlaps(A, B)	Returns true if geometry A and geometry B share space, but are not completely contained by each other
ST_Touches(A, B)	Returns true if geometry A and geometry B share space, but are not completely contained by each other
ST_Within(A, B)	Returns true if geometry A is within geometry B

Section 13: Spatial Joins Exercises

What subway station is in the 'Little Italy' neighborhood?

What are all the neighborhoods served by the 6 train?

The "nyc_subway_stations" table has a "routes" column with values like 'A,C,6' and '5,6,Q'.

After 9/11, the 'Battery Park' neighborhood was off limits for several days. How many people had to be evacuated?

The "nyc_census_blocks" table has a "popn_total" column.

What neighborhood has the highest population density (persons/km²)? The lowest?

There are 1,000,000 m² in a km².

Useful SQL

The pattern for a spatial join is commonly

```
SELECT a.field, b.field
FROM table_a AS a
JOIN table_b AS b
ON ST_Something(a.the_geom, b.the_geom)
WHERE a.field = 'SOMETHING';
```

For spatial joins that aggregate results over the whole set, the pattern is commonly

```
SELECT Sum(a.field), b.field
FROM table_a AS a
JOIN table_b AS b
ON ST_Something(a.the_geom, b.the_geom)
WHERE a.field = 'SOMETHING'
GROUP BY b.field;
```

Note the aggregate function around one term and the grouping on the other.

Useful Functions

strpos(str, char)	Returns the character position of the character in the string or 0 if the character is not in the string.
field LIKE '%thing%'	Returns true if 'thing' is appears within a string in the field
Sum(field)	An aggregate function that Returns the total value of the field over all records in the query.
ST_Area(geometry)	Returns the area of the geometry in square units
ST_Contains(A, B)	Returns true if geometry A contains geometry B
ST_Crosses(A, B)	Returns true if geometry A crosses geometry B
ST_Disjoint(A, B)	Returns true if the geometries do not "spatially intersect"
ST_Distance(A, B)	Returns the minimum distance between geometry A and geometry B
ST_DWithin(A, B, d)	Returns true if geometry A is distance or less from geometry B
ST_Equals(A, B)	Returns true if geometry A is the same as geometry B
ST_Intersects(A, B)	Returns true if geometry A intersects geometry B
ST_Overlaps(A, B)	Returns true if geometry A and geometry B share space, but are not completely contained by each other
ST_Touches(A, B)	Returns true if geometry A and geometry B share space, but are not completely contained by each other
ST_Within(A, B)	Returns true if geometry A is within geometry B

Section 16: Projection Exercises

What is the SRID of the nyc_streets table?

What projection does that SRID represent?

What is the total length of all streets in New York, as measured in UTM 18N?

See the question above!

What is the total length of all streets in New York, as measured in Stateplane Long Island (EPSG 2831)?

Why is the answer different from above?!

How many distinct streets cross the 74th meridian?

The data run from about -73.5 to -74.5 longitude and 40.5 to 40.9 latitude.
Streets can have multiple rows with the same name.

Useful Metadata

The “spatial_ref_sys” table contains all the information about SRID values in the database. It is user-configurable, but is loaded by default with all the values from the EPSG database.

srid	Unique identifier for this spatial reference system
auth_name	Authority that defined this system (usually EPSG)
auth_srid	Identifier used by the authority (note that this does not have to be the same as the value used for the internal database SRID)
srttext	“well known text” definition of the spatial reference system. The same format as used in “prj” files.
proj4text	Proj4 text definition of the spatial reference system. Used by the PostGIS re-projection engine in ST_Transform calls.

Useful Functions

DISTINCT	Returns only unique rows (all values the same)
Sum(field)	An aggregate function that Returns the total value of the field over all records in the query.
ST_Transform(A, srid)	Returns a geometry in the requested spatial reference system, transforming the coordinates of the geometry.
ST_SRID(A)	Returns the SRID value of the geometry
ST_SetSRID(A, srid)	Updates the SRID value of a geometry, does not alter the coordinates.
ST_GeomFromText(text, srid)	Create a new geometry from a well-known text form and set the SRID value at the same time.
ST_XMin(A)	Returns the minimum value of the X coordinate of a geometry.
ST_XMax(A)	Returns the maximum value of the X coordinate of a geometry.
ST_Intersects(A, B)	Returns true of geometry A and geometry B intersect, false if they are disjoint.

Section 17: Geography Exercises

How far is New York from Seattle? What are the units of the answer?

New York = POINT(-74.0064 40.7142) , Seattle = POINT(-122.3331 47.6097)

What is the total length of all streets in New York, calculated on the spheroid?

You can cast geometry to geography using the "::geography" syntax or the "geography(geometry)" function.

Does 'POINT(1 2.0001)' intersect with 'POLYGON((0 0, 0 2, 2 2, 2 0, 0 0))' in geography? In geometry? Why the difference?

ST_GeomFromText and ST_GeogFromText are helpful here.

Casting Types

You can convert values from one PostgreSQL type to another using the "::type" syntax.

```
-- Converts text to geometry
```

```
SELECT 'POINT(0 0)::geometry;
```

```
-- Converts text to geography to geometry
```

```
SELECT 'POINT(0 0)::geography::geometry
```

Useful Functions

Sum(field)	An aggregate function that Returns the total value of the field over all records in the query.
ST_Transform(geom, srid)	Returns a geometry in the requested spatial reference system, transforming the coordinates of the geometry.
ST_GeomFromText(text, srid)	Create a new geometry from a well-known text form and set the SRID value at the same time.
ST_Distance(geog1, geog2)	Returns the shortest distance in meters between two geographies.
ST_Length(geog)	Returns the length in meters of a geography.
ST_GeogFromText(text)	Create a new geography from a well-known text form.

Section 17: Geometry Construction Exercises

How many census blocks don't contain their own centroid?

Union all the census blocks into a single output. What kind of geometry is it?
How many parts does it have?

What is the area of a one unit buffer around the origin? How different is it from what you would expect? Why?

The value of π can be accessed with the `pi()` function.

There is a three-parameter variant of `ST_Buffer(geom, radius, n)` What do you think the third parameter controls?

The Brooklyn neighborhoods of 'Park Slope' and 'Carroll Gardens' are going to war! Construct a polygon delineating a 100 meter wide DMZ on the border between the neighborhoods. What is the area of the DMZ?

Hint. A self-join will be required. Buffer and then intersect.

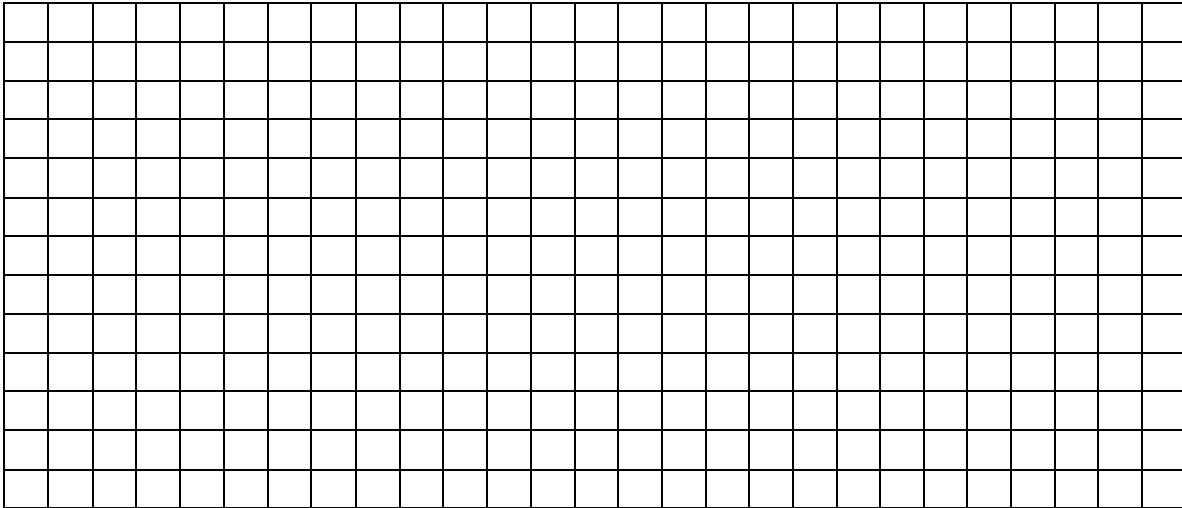
Useful Functions

Count(field)	An aggregate function that returns the number of rows in the query result set.
ST_Intersection(A, B)	Returns a geometry that covers the area shared by geometries A and B.
ST_Centroid(A)	Returns a point at the center of mass of geometry A.
ST_PointOnSurface(A)	Returns a point guaranteed to be within the boundary of geometry A.
ST_Union(A, B)	Returns a geometry that covers all the area covered by geometry A or geometry B.
ST_Contains(A, B)	Returns true if geometry A contains geometry B.
ST_Union([geometryset])	Aggregate function returns a geometry that covers all the area covered by all elements of the input geometry set.
ST_GeometryType(A)	Returns the geometry type of geometry A.
ST_NumGeometries(A)	Returns the number of sub-geometries in a geometry collection.
Pi()	Returns the value of pi.
ST_Buffer(A, d)	Returns a polygonal geometry that covers all the area within a radius d of geometry A and all the area within A itself.

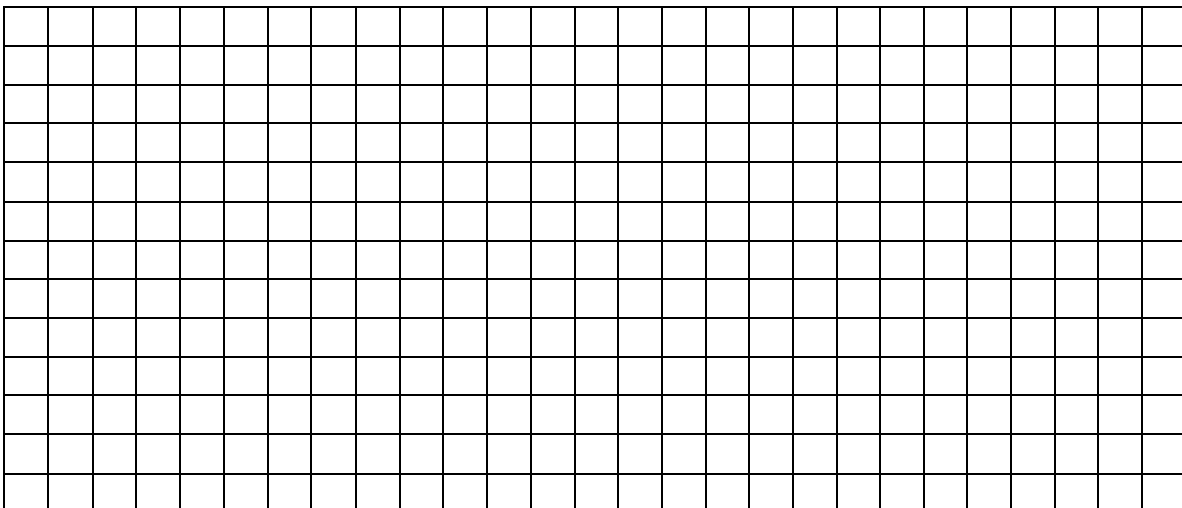
Section 17: Validity Exercises

Create an invalid geometry. Verify it is invalid and why the system thinks it is invalid.

The rules: Interior rings inside exterior, no ring intersections except at a single point, no ring self-intersections, no multi-polygons with intersecting parts.
Here is some graph paper to make it easier!



Try to repair your invalid geometry. What is the result?

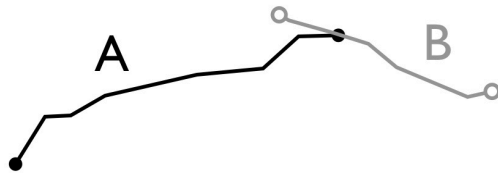


Useful Functions

ST_GeomFromText(wkt, srid)	Create a geometry from a well-known text representation.
ST_IsValid(geometry)	Returns true if the geometry is constructed according to validity rules.
ST_IsValidReason(geometry)	Returns a string describing the invalidity of a geometry or NULL if the geometry is valid.
ST_IsValidDetail(geometry)	Returns a composite object including a validity string and a point location where the invalidity occurs, or NULL if the geometry is valid.
ST_Buffer(geometry, 0)	Rebuilds (often) a valid version of an invalid input polygon.
ST_Perimeter(geometry)	Returns the total length of line work in a polygon boundary, unaffected by invalidities.
ST_Area(geometry)	Returns the area of a geometry, may be affected by invalidity (in particular inverted ring sections).

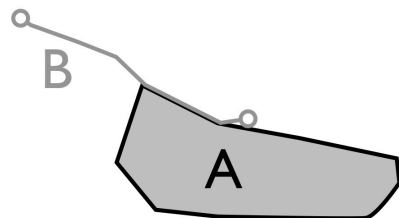
Section 17: DE9IM Exercises

Fill in the DE9IM matrix for this geometry interaction.



		B		
		I	B	E
A	I			
	B			
	E			

Fill in the DE9IM matrix for this geometry interaction.



		B		
		I	B	E
A	I			
	B			
	E			

Section 1: Simple SELECT Exercises

Which actors are named 'GRETA'?

How long can I take out 'CHICAGO NORTH' for? What will it cost me?

What 'G' rated films are exactly 48 minutes long?

I'm looking for a short (no more than 1 hour) cheap movie (less than \$1) about a dentist. What can you offer me?

I'm looking for an 'Emotional Drama'. It's about either a 'Squirrel' or a 'Dog', I think. Do you have that?

Data Dictionary

Useful columns in the “**film**” table:

film_id	A unique id for every film
title	Title of the film
description	Description of the film
release_year	Year of release
rental_duration	How long (days) is film allowed to be rented out for?
rental_rate	How much does renting this film cost?
length	How long (minutes) is this film?
replacement_cost	How much will buying a new copy cost?

Useful columns in the “**actor**” table:

actor_id	A unique id for every actor
first_name	First name of the actor
last_name	Last name of the actor

Useful Functions

LIKE	Operator for WHERE clause, uses '%' as a global match character.
ILIKE	Case insensitive version of LIKE.
=	Operator for WHERE clause, tests strict equality.
>	Operator for WHERE clause, tests greater-than relationship.
<	Operator for WHERE clause, tests less-than relationship.
~	Regular expression match.
~*	Case insensitive regular expression match.

Section 2: Aggregates and Ordering

What is the maximum replacement cost of a film?

What is average film rental rate, summarized by the allowed rental durations?

What 10 films about a 'Cat' are the cheapest to rent?

(Hint: order, limit)

How many films have a 'Monk' and a 'Dog' in them?

How strong is the correlation between movie length and rental duration?

Data Dictionary

Useful columns in the “**film**” table:

film_id	A unique id for every film
title	Title of the film
description	Description of the film
release_year	Year of release
rental_duration	How long (days) is film allowed to be rented out for?
rental_rate	How much does renting this film cost?
length	How long (minutes) is this film?
replacement_cost	How much will buying a new copy cost?

Useful Functions

Sum([field])	An aggregate function that returns the total value of the field over all records in the query.
LIMIT n	Restrict the query to return only the first “n” rows.
ORDER BY [field]	Return the query in order sorted by the field.
ORDER BY [field] DESC ORDER BY [field] ASC	Return the query in descending/ascending order sorted by the field.
GROUP BY [field]	Return aggregate functions using categories defined by the field
Avg ([field])	Aggregate that returns the average value of the field over all records in the group
StdDev ([field])	Aggregate that returns the standard deviation value of the field over all records in the group
Count ([field])	Aggregate that returns the number of records in the group
Min ([field])	Aggregate that returns the minimum value of the field over all records in the group
Max ([field])	Aggregate that returns the maximum value of the field over all records in the group
String_Agg([field],[string])	Aggregate that returns all values of the field concatenated into a single string using the provided string as a separator
ST_Corr([field1], [field2])	Aggregate that returns the correlations between the values in the provided fields, over the query groups

Section 3: Joins

Who else acted in 'MEMENTO ZOOLANDER' with Kirsten Akroyd?

How many films are in each category?

Do any of the films in the 'Music' category have 'Music' in their description?

What is the most common category of movie made by actresses named 'Julia'?

How much revenue did the movie store generate in January of 2007?

How much of that was late fees?

For SQL experts only! Get out your ER diagram!

Useful Functions

JOIN [column] ON [condition]	Join tables using a true/false condition (this will be important when we get to spatial and don't have foreign keys to work with)
JOIN [column] USING ([key])	Joins tables using a key that has same column name in both tables.
Sum([field])	Aggregate function returning sum of all inputs.

Useful columns in the “**film**” table:

film_id	A unique id for every film
title	Title of the film
release_year	Year of release
rental_duration	How long (days) is film allowed to be rented out for?
rental_rate	How much does renting this film cost?
length	How long (minutes) is this film?

Useful columns in the “**actor**” table:

actor_id	A unique id for every actor
first_name	First name of the actor
last_name	Last name of the actor

Useful columns in the “**film_actor**” table:

actor_id	Actor associated with film
film_id	Film associated with actor

Useful columns in the “**film_category**” table:

category_id	Category associated with film
film_id	Film associated with category

Useful columns in the “**category**” table:

category_id	Category unique id
name	Human readable category name

Section 4: CRUD

Add a new actor table entry for your favorite thespian.

Change your thespian's last name to 'SHATNER'.

Re-rate all the 'G' films as 'PG'.

Delete payment 16076.

(Try to) delete film 5. Why doesn't this work?

Useful Functions

INSERT INTO [table] ([columns]) VALUES ([values]) RETURNING [column]	Add a record to a table
UPDATE [table] SET [column] = [value] WHERE [condition]	Alter existing record(s) in a table.
DELETE FROM [table] WHERE [condition]	Delete record(s) from a table.

Useful columns in the “**film**” table:

film_id	A unique id for every film
title	Title of the film
release_year	Year of release
rental_duration	How long (days) is film allowed to be rented out for?
rating	MPAA rating
length	How long (minutes) is this film?

Useful columns in the “**actor**” table:

actor_id	A unique id for every actor
first_name	First name of the actor
last_name	Last name of the actor

Useful columns in the “**payment**” table:

payment_id	A unique id for every payment
amount	Size of payment (dollars)